Zadania EXTRA dla studentów   
(zaawansowana zmiana historii repozytorium)

Przygotowanie:

* Stworzyć forka repozytorium: <https://github.com/kneefer/RepozytoriaKoduExtra> (przycisk **Fork**   
  w prawym górnym rogu)
* Sklonować sforkowane repozytorium lokalnie (za pomocą komendy git clone lub Visual Studio GIT Extensions) [https://github.com/<twojuzytkownik>/RepozytoriaKoduExtra](https://github.com/%3ctwojuzytkownik%3e/RepozytoriaKoduExtra)
* Opcjonalnie – wykonać polecenie git config --global core.editor "nano" – zmienia konsolowy edytor tekstowy, którego GIT będzie używał na **nano**. Bardziej zaawansowani mogą   
  z tego zrezygnować (wtedy GIT będzie używał domyślnego edytora **vi**)
* Repozytorium zawiera 5 przykładowych commitów (o nazwach które sugerują kolejność utworzenia ich w repozytorium):
  + 1. Added file1.txt
  + 2. Added file2.txt
  + 3. Added file3.txt
  + 4. Removed file1.txt, added file.4.txt
  + 5. Updated file2.txt

Scalanie commitów:

* Stworzyć nowego brancha z mastera o nazwie squash i na nim pracować
* Użyć interaktywnego rebase’a, aby zmienić historię i scalić commity **#2-#4**  
  git rebase –i HEAD~4 (**HEAD** to wskaźnik na commit **#5** – najnowszy, **HEAD~1** na **#4**, itd. Polecenie rebase przyjmuje wskaźnik na RODZICA commita od którego ma rozpocząc proces)
* W pierwszym kroku należy na odpowiednich commitach zmienić **pick** na **squash** (zgodnie z instrukcją, która się wyświetli) i zapisać plik
* W drugim kroku należy zmodyfikować plik tak, aby ustalić nową nazwę dla scalonego commita, zawierającego zmiany z commitów **#2-#4** (zgodnie z instrukcją która się wyświetli) i zapisać plik
* Za pomocą polecenia git push opublikować brancha na zdalnym repozytorium

Rozbijanie commitów:

* Stworzyć nowego brancha z mastera o nazwie edit i na nim pracować
* Użyć interaktywnego rebase’a, aby zmienić historię i rozbić commit **#4** na dwa osobne  
  git rebase –i HEAD~2
* W pierwszym kroku należy na commicie **#4** zmienić **pick** na **edit** (zgodnie z instrukcją, która się wyświetli) i zapisać plik
* W drugim kroku należy wykorzystać polecenie git reset aby zresetować zmiany z ostatniego commita (w wyniku rebase’a i zaznaczenia akcji dla commita 4 jako **edit,** HEAD będzie wskazywał na commit **#4**)
* Następnie należy za pomocą poleceń git add oraz git commit zastage’ować oraz zacommitować odpowiednie zmiany tak, aby commit **#4** został rozbity na commit usuwający plik file1.txt oraz commit dodający plik file.4.txt.
* Kontynuować (zakończyć) proces rebase’a wykonując polecenie git rebase –continue
* Za pomocą polecenia git push opublikować brancha na zdalnym repozytorium